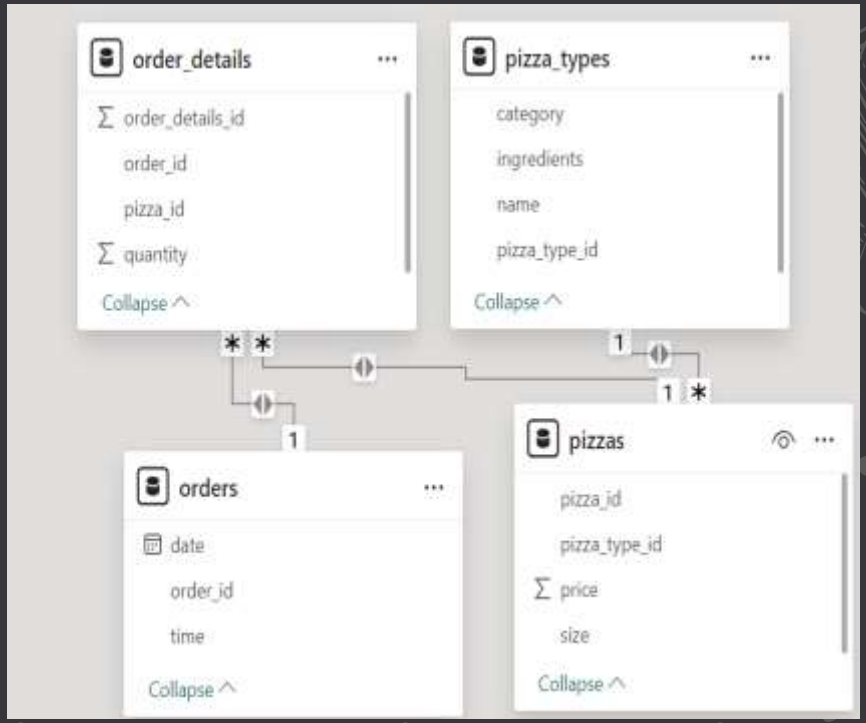
# SQL PROJECT ON PIZZA SALES

# HELLO !

Pizza is one of the most popular and widely consumed fast foods globally, known for its versatility and appeal across different cultures and demographics. Understanding pizza sales involves analyzing various factors such as customer preferences, seasonal trends, pricing strategies, and regional influences.

# Dataset Schema

The pizza sales dataset provides insights into the performance of a pizza business, capturing details like order dates, customer preferences, sales amount , and popular menu items. It helps analyze trends, such as peak sales periods, best-selling pizzas, and regional demand, enabling businesses to make data-driven decisions and optimize their operations.



# Analysis

Basic:

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.

Intermediate:

1. Join the necessary tables to find the total quantity of each pizza category ordered.
2. Determine the distribution of orders by hour of the day.
3. Join relevant tables to find the category-wise distribution of pizzas.
4. Group the orders by date and calculate the average number of pizzas ordered per day.
5. Determine the top 3 most ordered pizza types based on revenue.

Advanced:

1. Calculate the percentage contribution of each pizza type to total revenue.
2. Analyze the cumulative revenue generated over time.
3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

# Retrieve the total number of orders placed.

```
SELECT
        COUNT(order_id) AS TOTAL_ORDERS
FROM
      orders$;
```

Results | Messages

| | TOTAL_ORDERS |
|---|---|
| 1 | 21350 |

# Calculate the total revenue generated from pizza sales.

```
SELECT
      ROUND(SUM(pizzas$.price*order_details$.quantity),
      2) AS TOTAL_REVENUE
FROM
          pizzas$
         INNER JOIN
    order_details$ ON order_details$.pizza_id=pizzas$.pizza_id
```

Results | Messages

| | TOTAL_REVENUE |
|---|---|
| 1 | 817860.05 |

# Identify the highest-priced pizza.

```sql
SELECT
      TOP 1 pizza_types$.name AS PIZZA_NAME,MAX(price) AS HIGHEST_PRICED_PIZZA
FROM
        pizzas$
         INNER JOIN
         pizza_types$ ON pizza_types$.pizza_type_id=pizzas$.pizza_type_id
GROUP BY pizza_types$.name
ORDER BY HIGHEST_PRICED_PIZZA DESC
```

Results | Messages

| | PIZZA_NAME | HIGHEST_PRICED_PIZZA |
|---|---|---|
| 1 | The Greek Pizza | 35.95 |

# Identify the most common pizza size ordered.

```
SELECT
      TOP 1 SIZE,
      COUNT(order_details$.order_id) AS MOST_COMMON_PIZZA_SIZE_ORDERED
FROM
       pizzas$
        INNER JOIN
       order_details$ ON
       pizzas$.pizza_id=order_details$.pizza_id
GROUP BY size
ORDER BY MOST_COMMON_PIZZA_SIZE_ORDERED DESC
```

Results | Messages

| | SIZE | MOST_COMMON_PIZZA_SIZE_ORDERED |
|---|---|---|
| 1 | L | 18526 |

# List the top 5 most ordered pizza types along with their quantities.

```
SELECT
      TOP 5 pizza_types$.name AS PIZZA_NAME,
      SUM(order_details$.quantity) AS QUANTITY
FROM
        pizza_types$
        INNER JOIN
        pizzas$ ON pizza_types$.pizza_type_id=pizzas$.pizza_type_id
        INNER JOIN
        order_details$ ON order_details$.pizza_id=pizzas$.pizza_id
GROUP BY pizza_types$.name
ORDER BY QUANTITY DESC
```

Results | Messages

| | PIZZA_NAME | QUANTITY |
|---|---|---|
| 1 | The Classic Deluxe Pizza | 2453 |
| 2 | The Barbecue Chicken Pizza | 2432 |
| 3 | The Hawaiian Pizza | 2422 |
| 4 | The Pepperoni Pizza | 2418 |
| 5 | The Thai Chicken Pizza | 2371 |

# Join the necessary tables to find the total quantity of each pizza category ordered.

```
SELECT
      pizza_types$.category,
      SUM(order_details$.quantity) AS TOTAL_QUANTITY
FROM
       pizza_types$
       INNER JOIN
       pizzas$ ON pizza_types$.pizza_type_id=pizzas$.pizza_type_id
       INNER JOIN
       order_details$ ON order_details$.pizza_id=pizzas$.pizza_id
GROUP BY pizza_types$.category
ORDER BY TOTAL_QUANTITY
```

Results | Messages

| | category | TOTAL_QUANTITY |
|---|---|---|
| 1 | Chicken | 11050 |
| 2 | Veggie | 11649 |
| 3 | Supreme | 11987 |
| 4 | Classic | 14888 |

# Determine the distribution of orders by hour of the day.

```
SELECT
        DATEPART(HOUR,time) as hour,
        count(order_id) as order_count
FROM
    orders$
group by DATEPART(HOUR,time)
order by hour
```

Results | Messages

| | hour | order_count |
|---|---|---|
| 1 | 9 | 1 |
| 2 | 10 | 8 |
| 3 | 11 | 1231 |
| 4 | 12 | 2520 |
| 5 | 13 | 2455 |
| 6 | 14 | 1472 |
| 7 | 15 | 1468 |
| 8 | 16 | 1920 |
| 9 | 17 | 2336 |
| 10 | 18 | 2399 |
| 11 | 19 | 2009 |
| 12 | 20 | 1642 |
| 13 | 21 | 1198 |
| 14 | 22 | 663 |
| 15 | 23 | 28 |

# Join relevant tables to find the category-wise distribution of pizzas.

```
select
      category,
      COUNT(name) as distribution_of_pizzas
from pizza_types$
group by category
```

Results | Messages

| | category | distribution_of_pizzas |
|---|---|---|
| 1 | Chicken | 6 |
| 2 | Classic | 8 |
| 3 | Supreme | 9 |
| 4 | Veggie | 9 |

# Group the orders by date and calculate the average number of pizzas ordered per day.

```
select
	round(avg(total_quantity),0) as avg_orders_per_day
from
	(select
		CONVERT(varchar,date,111) as date,
		SUM(quantity) as total_quantity
	  from
		order_details$
		 inner join
		orders$ on order_details$.order_id=orders$.order_id
	 group by CONVERT(varchar,date,111)
	 ) as orders_quantity;
```

Results | Messages

| | avg_orders_per_day |
|---|---|
| 1 | 138 |

# Determine the top 3 most ordered pizza types based on revenue.

```sql
select
	top 3 pizza_types$.name as pizza_name,
	SUM(quantity*price) as total_revenue
from
	order_details$
	   inner join
	pizzas$ on order_details$.pizza_id=pizzas$.pizza_id
	  inner join
	pizza_types$ on pizza_types$.pizza_type_id=pizzas$.pizza_type_id
group by pizza_types$.name
order by total_revenue desc
```

Results | Messages

| | pizza_name | total_revenue |
|---|---|---|
| 1 | The Thai Chicken Pizza | 43434.25 |
| 2 | The Barbecue Chicken Pizza | 42768 |
| 3 | The California Chicken Pizza | 41409.5 |

# Calculate the percentage contribution of each pizza type to total revenue.

```
select
	pizza_types$.category,
	round(
		(SUM(order_details$.quantity*pizzas$.price)/
			(SELECT
				ROUND(SUM(pizzas$.price*order_details$.quantity),2) AS TOTAL_REVENUE
			FROM
				pizzas$
				INNER JOIN
				order_details$
				ON order_details$.pizza_id=pizzas$.pizza_id))*100,2) as percentage_contribution
	from
		pizza_types$
		inner join
		pizzas$ on pizza_types$.pizza_type_id=pizzas$.pizza_type_id
		inner join
		order_details$ on order_details$.pizza_id=pizzas$.pizza_id
group by pizza_types$.category
```

Results | Messages

| | category | percentage_contribution |
|---|---|---|
| 1 | Chicken | 23.96 |
| 2 | Supreme | 25.46 |
| 3 | Classic | 26.91 |
| 4 | Veggie | 23.68 |

# Analyze the cumulative revenue generated over time.

```
select
        date1,sum(revenue) over(order by date1) as cumulative_revenue
        from
         (select
                CONVERT(varchar,date,111) as date1,
                SUM(order_details$.quantity*pizzas$.price) as revenue
         from
               orders$
                 inner join
               order_details$ on order_details$.order_id=orders$.order_id
                 inner join pizzas$ on pizzas$.pizza_id=order_details$.pizza_id
           group by orders$.date) as sales;
```

Results | Messages

| | date1 | cumulative_revenue |
|---|---|---|
| 1 | 2015/01/01 | 2713.85 |
| 2 | 2015/01/02 | 5445.75 |
| 3 | 2015/01/03 | 8108.15 |
| 4 | 2015/01/04 | 9863.6 |
| 5 | 2015/01/05 | 11929.55 |
| 6 | 2015/01/06 | 14358.5 |
| 7 | 2015/01/07 | 16560.7 |
| 8 | 2015/01/08 | 19399.05 |

# Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
select
        Name,revenue
from
       (select Name,category,revenue,
              RANK() over(partition by category order by revenue desc) as rank
       from
            (select pizza_types$.name as Name,
                    pizza_types$.category as category,
                    SUM(order_details$.quantity*pizzas$.price) as revenue
            from
                   pizza_types$
                    inner join
                    pizzas$ on pizza_types$.pizza_type_id=pizzas$.pizza_type_id
                      inner join
                    order_details$ on order_details$.pizza_id=pizzas$.pizza_id
                group by pizza_types$.category,pizza_types$.name
                ) as a
        ) as b
where rank<=3;
```

Results | Messages

| | Name | revenue |
|---|---|---|
| 1 | The Thai Chicken Pizza | 43434.25 |
| 2 | The Barbecue Chicke... | 42768 |
| 3 | The California Chicke... | 41409.5 |
| 4 | The Classic Deluxe Pi... | 38180.5 |
| 5 | The Hawaiian Pizza | 32273.25 |
| 6 | The Pepperoni Pizza | 30161.75 |
| 7 | The Spicy Italian Pizza | 34831.25 |
| 8 | The Italian Supreme ... | 33476.75 |
| 9 | The Sicilian Pizza | 30940.5 |
| 10 | The Four Cheese Pizza | 32265.... |
| 11 | The Mexicana Pizza | 26780.75 |
| 12 | The Five Cheese Pizza | 26066.5 |